Technical Report 484

DESIGN RECOMMENDATIONS FOR QUERY LANGUAGES

S.L. Ehrenreich

HUMAN FACTORS TECHNICAL AREA

U. S. Army
Research Institute for the Behavioral and Social Sciences

September 1980

U. S. ARMY RESEARCH INSTITUTE
FOR THE BEHAVIORAL AND SOCIAL SCIENCES

A Field Operating Agency under the Jurisdiction of the

Deputy Chief of Staff for Personnel

L. NEALE COSBY
Colonel, IN
Commander

JOSEPH ZEIDNER
Technical Director

NOTICES

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER TR 484	2. GOVT ACCESSION NO. [illegible]	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) DESIGN RECOMMENDATIONS FOR QUERY LANGUAGES		5. TYPE OF REPORT & PERIOD COVERED -- 6. PERFORMING ORG. REPORT NUMBER --
7. AUTHOR(s) S. L. Ehrenreich		8. CONTRACT OR GRANT NUMBER(s) --
9. PERFORMING ORGANIZATION NAME AND ADDRESS U.S. Army Research Institute for the Behavioral and Social Sciences 5001 Eisenhower Ave., Alexandria, VA 22333		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 2Q162717A790
11. CONTROLLING OFFICE NAME AND ADDRESS Deputy Chief of Staff Personnel Washington, DC 20310		12. REPORT DATE September 1980 13. NUMBER OF PAGES 39
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) --		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE --

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

--

18. SUPPLEMENTARY NOTES

--

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Query languages	Man-computer interaction
Computer languages	Natural computer languages
Man-computer dialogues	Formal computer languages

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The existing human factors literature on query languages is both sparse and scattered. This paper seeks to collect and review that literature. The first section of the paper introduces the subject of query languages. In the second and third sections, the topics of natural and formal query languages are respectively discussed. These two types of query languages are reviewed with the objective of determining their potential for expanding the population of computer users. The fourth section considers some general issues pertinent to both types of query languages. These issues include the

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. (continued)

ability of people to deal with logical quantifiers, the user's concept of data organization, mixed initiative dialogues, and the use of abbreviations. Methods for experimentally evaluating specific query language features and research on person-to-person communication are also discussed here. To focus the findings reported in the preceding sections, the fifth section summarizes the implications of the research performed to date. Next, the sixth section presents possible new research which would be of value to the designers of Army tactical information systems. The paper concludes with two appendixes. Appendix A discusses human factors review papers concerned with the design of interactive systems. Appendix B presents a compendium of design recommendations directed towards the system designer.

Technical Report 484

DESIGN RECOMMENDATIONS FOR QUERY LANGUAGES

S.L. Ehrenreich

Submitted by:
Stanley M. Halpin, Acting Chief
HUMAN FACTORS TECHNICAL AREA

Approved by:

Edgar M. Johnson
ORGANIZATIONS AND SYSTEMS
RESEARCH LABORATORY

U.S. ARMY RESEARCH INSTITUTE FOR THE BEHAVIORAL AND SOCIAL SCIENCES
5001 Eisenhower Avenue, Alexandria, Virginia 22333

Office, Deputy Chief of Staff for Personnel
Department of the Army

September 1980

Army Project Number
2Q162717A790

Human Performance
Effectiveness

ARI Research Reports and Technical Reports are intended for sponsors of R&D tasks and for other research and military agencies. Any findings ready for implementation at the time of publication are presented in the last part of the Brief. Upon completion of a major phase of the task, formal recommendations for official action normally are conveyed to appropriate military agencies by briefing or Disposition Form.

FOREWORD

The Human Factors Technical Area of the Army Research Institute (ARI) is concerned with the demands of increasingly complex battlefield systems that are used to acquire, transmit, process, disseminate, and utilize information. This increased complexity places greater demands upon the operator interacting with the machine system. Research in this area is focused on human performance problems related to interactions within command and control centers as well as on issues of systems development. Such research is concerned with software development, topographic products and procedures, tactical symbology, user-oriented systems, information management, staff operations and procedures, decision support, and sensor systems integration and utilization.

An issue of special concern within the area of user-oriented systems is simplifying the user-computer dialogue. The increasing utilizing of computers in battlefield and other Army systems has created a demand for a large number of competent computer operators. In order to satisfy this need, the language used to communicate with computers must be made simpler, easier to learn and less prone to errors. A variety of dialogue languages are available for user-computer communication. The present publication reviews the human factors research concerned with query languages and their potential for simplifying user-computer transactions. Existing research reports are reviewed for their operational implications and for their implications with regard to future research needs.

Research in user-oriented systems is conducted as an in-house effort augmented through contracts. This report resulted from an in-house research effort responsive to requirements of Army Project 2Q162717A790. Special requirements are contained in Thrust 4, Work Unit 002, "Design and Evaluation of User-System Transactions."

Joseph Zeidner

JOSEPH ZEIDNER
Technical Director

DESIGN RECOMMENDATIONS FOR QUERY LANGUAGES

BRIEF

Requirements:

To improve the design of query languages by making them simpler to use, easier to learn and less prone to user error.

Procedure:

The existing human factors literature on query languages is both sparse and scattered. This paper seeks to collect and review that literature. The first section of the paper introduces the subject of query languages. In the second and third sections, the topics of natural and formal query languages are respectively discussed. These two types of query languages are reviewed with the objective of determining their potential for expanding the population of computer users. The fourth section considers some general issues pertinent to both types of query languages. These issues include the ability of people to deal with logical quantifiers, the user's concept of data organization, mixed initiative dialogues, and the use of abbreviations. Methods for experimentally evaluating specific query language features and research on person-to-person communication are also discussed here. To focus the findings reported in the preceding sections, the fifth section summarizes the implications of the research performed to date. Next, the sixth section presents possible new research which would be of value to the designers of Army tactical information systems. The paper concludes with two appendixes. Appendix A discusses human factors review papers concerned with the design of interactive systems. Appendix B presents a compendium of design recommendations directed towards the system designer.

Findings:

Much work remains to be done in setting up design guidelines for query languages. The research guidance that is available in the human factors literature is summarized at the end of this paper. In addition, more specific design guidelines are presented in Appendix B.

Utilization of Findings:

This report brings together the principle results of research efforts in the area of query languages. It provides interested system proponents and developers with recommendations and guidance for improving the dialogue between users and their computers.

DESIGN RECOMMENDATIONS FOR QUERY LANGUAGES

CONTENTS

DESIGN RECOMMENDATIONS FOR QUERY LANGUAGES

INTRODUCTION

The United States Army continues to introduce computers into more and more areas of its operations. These include the areas of data processing for combat as well as for noncombat situations. For example, more than 90 separate automated battlefield systems are either under development or in production. These systems will increase the power of the command staff to integrate and retrieve important intelligence, logistic, and other battlefield information. To effectively utilize this increase in power, there will have to be enough individuals capable of operating the new systems. All of these users will not be highly skilled and well trained computer technicians and programmers. For the U.S. Army, the shortage in such skilled personnel is especially acute. Therefore, it is up to the system designer to simplify the techniques for user-computer transactions and thus assure that the number of potential, competent computer users increases along with the expanding need. (For brevity, the term "user" rather than the term "user/operator" will be used in this paper.)

A systems component which is a prime candidate for simplification is the user-computer dialogue. Several types of interactive dialogue can be incorporated into a computer system. They include the following:

1. Question-and-Answer Dialogue--The computer asks a question which requires a "yes," "no," or "don't know" answer from the user. The user's response causes the computer system to determine which question should be asked next. The succession of questions and answers guides the program to the action that the user desires. (This action can involve retrieving information, manipulating information, or initiating a physical action.)

2. Form Filling Dialogue--The computer presents the user with a standard text. At a number of points, the text requests specific information which the user types in. This information guides the computer in the performance of the desired task. (The distinction between form filling and question-and-answer dialogue can become obscure under some circumstances.)

3. Menu Selection Dialogue--The computer asks a question of the user and also presents a list of possible answers. The answers chosen by the user determine what task will be performed.

4. Query Language Dialogue--Unlike the other types of interactive dialogue, query languages do not require that the computer guide the dialogue. A query language is a set of syntactic and lexical rules (i.e., language) with which the user can question (i.e., query) the computer. Query languages belong to the class of computer languages commonly referred to as "nonprocedural" or "very high level" (Leavenworth & Sammet, 1974). In nonprocedural languages, the user declares what the program is to accomplish without stating how it is to be accomplished (i.e., without providing a procedure). Query languages can be characterized by their syntax and vocabulary. In a natural query language,

the syntax and vocabulary of the query language closely resembles that of English (which we will assume to be the user's natural language). On the other hand, the syntax and vocabulary of a formal query language is highly constrained and has little resemblance to English. Below are examples of a statement written both in natural and in a formal query language.

Natural: Find the names of all of the employees in department number 50.

Formal (written in GIM II): FROM EMP WITH DEPTNO EQ "50" LIST NAME #

A number of properties distinguishes query languages from other classes of computer languages. For instance, the nonprocedural aspect of query languages is useful in distinguishing them from programming languages. Another distinction between these two classes of languages is that each query statement is executed by the system upon entry, whereas the execution of a programming statement is delayed until the total program has been entered. Differentiation can also be made between query languages and command languages. Like query languages, command languages are nonprocedural and each statement is executed immediately. However, command languages are not primary tools used for the creation of problem-solving algorithms. Instead, they are secondary tools (e.g., job control languages, text editors) used to execute programs conveniently (Gram & Hertweck, 1975). The above distinctions (between types of languages and the categorization of dialogue types) are not intended as hard and fast definitions. Instead, they are being stated so that the reader will be cognizant of the author's perspective.

The existing human factors literature on query languages is both sparse and scattered. This paper seeks to collect and review that literature. The present section has introduced the subject of query languages. In the second and third sections, the topics of natural and formal query languages will be respectively discussed. These two types of query languages are reviewed with the objective of determining their potential for expanding the population of computer users. The fourth section considers some general issues pertinent to both types of query languages. These issues include the ability of people to deal with logical quantifiers, the user's concept of data organization, mixed initiative dialogues, and the use of abbreviations. Methods for experimentally evaluating specific query language features and research on person-to-person communication are also discussed here. To focus the findings reported in the preceding sections, the fifth section summarizes the implications of the research performed to date. Next, the sixth section presents possible new research which would be of value to the designers of Army tactical information systems. The paper concludes with two appendixes. Appendix A discusses human factors review papers concerned with the design of interactive systems. Appendix B presents a compendium of design recommendations directed towards the system designer.

In reading this article, the reader should keep in mind that most of the research done on human factors in query languages has been of limited scope. The reader must beware of over-generalizing any of the findings. Still, in the absence of anything more substantial, these findings do represent the best wisdom to date.

NATURAL QUERY LANGUAGES

A prime obstacle to the widespread use of computers is the need to learn a special language for communicating with the machine. One means for removing this barrier is to have user-computer communication occur via natural language. "We will say that a system understands natural language if a user who knows what the system is capable of but who has not specifically trained in the system's input language (i.e., its domain of competence) can phrase an input to the system and, possibly after some clarifying dialogue . . . have his input satisfactorily handled." (Shapiro & Kwasny, 1975, p. 459).

Success has been achieved in developing natural query languages which satisfy this criterion. Waltz (1976) reports on some of these languages, including his own project PLANES. PLANES is a natural query language capable of answering questions relating to aircraft flight and maintenance records. Below is a sample dialogue between a user and a computer (Waltz, 1976, p. 21):

User: How many Phantoms required unscheduled maintenance in April 1975?

PLANES: Thirty Phantoms. Do you want to see them?

User: No. Did any of these require radar maintenance?

PLANES: Yes, three of them. Do you want to see them?

User: Yes.

PLANES: Phantom 159829 required a new radar screen.....Phantom 158336 crashed as a result of radar failure.

User: Did any other planes crash?

PLANES: I assume you mean during April 1975. Two other planes crashed: a Phantom and a P3.

User: Is it true that in both cases and April flights occurred while on the fuselage?

PLANES: I cannot understand your question. It is nongrammatical and meaningless.

User: Is it true that both Phantoms were fron the same squadron?

PLANES: I assume you mean "from" instead of "fron." Yes both planes were from the same squadron.

Heidorn (1976) reports that natural query language is also being used by systems which perform automatic programming. Such systems query the user in English and formulate a computer program based on the responses. (This should not be confused with efforts to create programming languages which are English in style.) Other successful natural languages projects are reported by Martin (1973) and Petrick (1976).

Despite the success reported above, the technology of natural query languages has not progressed to the point of commercial or military application. The natural languages developed to date are restricted in versatility and are geared to highly specific subject matters. Even within the topics for which languages have been developed, they are limited in linguistic capability (Petrick, 1976). They cannot handle a large variety of syntactic structures and they have limited vocabularies. Commercially, they are expensive because of the large memory they require for operation. Also, as will be discussed below, some researchers feel that natural query language is a poor medium for user-computer dialogues.

Protocols and Restricted Syntax

Still, simple forms of natural languages are feasible and potentially useful both militarily and commercially. One way to achieve simplicity within a natural language system is to restrict the syntax and vocabulary permitted by the system. Gould, Lewis, and Becker (1976) investigated the ease and accuracy with which participants who were nonprogrammers could write protocols using a restricted English syntax. The participants were also tested on their ability to comprehend such protocols. (Strictly speaking this experiment is concerned with natural language programming and not natural query language. Still, its findings are relevant to the latter issue.) In the experiment, participants were shown figures made up of either: (1) colored blocks, or (2) typed arrays of X's and blanks. The participants task was to either: (1) describe the scene, or (2) write a procedure for reconstructing it. In one condition, participants were provided with a restricted, natural language syntax for writing the protocols. In the second condition, no syntactic restrictions were placed on the participants. (The restricted syntax that was studied in this experiment was very simple. Care should be taken in generalizing these results.) For the experimental condition where the participants saw colored blocks, the syntax was as follows:

```
START WITH______________________
               (block color)

PUT_____________________   ____________________   ___________________
       (block color)       (spatial relation,       (block color)
                           e.g., "to the right
                           of," "above")
```

For the experimental condition where the participant saw typed arrays, the permitted syntax consisted of the single statement:

```
HIT___________________   ____________________
     (key, i.e., X,        (times, e.g., 3)
     space, return)
```

Participants found it just as easy to work with a restricted syntax as with an unrestricted one. The protocols that they produced when working with the restricted syntax were less ambiguous in their description of the scene and took no longer to prepare. Gould et al. also examined the relative ease with which the participants wrote protocols on how to construct the scene ("procedural protocols") as opposed to purely describing the scene ("description protocols"). The two types of protocols produced equally unambiguous descriptions of the

stimulus scene (i.e., were equally consistent with it). When participants prepared protocols under "neutral" constraint (i.e., without being instructed on the form that the protocol should take), they tended to produce more procedural protocols than description protocols. Gould et al. cautions against overgeneralizing their limited set of results. However, they point out that their experiment indicates that there is no "natural" form of expression for the design of a query language. Instead, people are flexible, capable of working with a well designed restricted syntax, and are able to prepare procedural instructions and do not naturally tend to pure descriptives.

Restricted Vocabulary

Kelly (1975) investigated the effect of restricting vocabulary size on the ability of people to communicate. In Kelly's experiment, individual college participants were placed in adjacent rooms and communicated with each other through teletype terminals. A pair of participants would be assigned a problem to solve (e.g., arrange a college course schedule given certain preconditions). Each participant in the pair was given half of the information required to solve the problem. Participants could communicate with each other under one of three vocabulary restrictions: (1) a vocabulary of 300 predefined words, (2) a vocabulary of 500 predefined words, or (3) no restrictions on vocabulary. The teletype interface between the participants was programmed so as to allow only the permissible vocabulary words to be used. Kelly found that vocabulary size had no effect on any measures of performance. This included both the time required and the accuracy with which the problems were solved. However, participants working under the limited vocabulary conditions did exhibit both annoyance and frustration with the system.

Kelly's experiment indicates that people can communicate within the confines of a limited-vocabulary language. This in turn increases the feasibility of creating an economical, English-based query language. However, there are some difficulties in attempting to generalize Kelly's results to communication between user and computer. Although Kelly provided his participants with a restricted vocabulary of only 300 words, the participants, in fact, had more than 300 different semantic entities available to them. Consider for example the word TIME which was one of the entries in the restricted vocabulary. This word has a multitude of meanings (e.g., time for reading, times are hard, doing time, getting paid time-and-a-half, learning to play piano in time). The single word TIME thus taps into many different semantic entries in the individual's lexicon. In a computer language system, each of these entries would have to be represented individually at a different memory address. Thus, a computer would require many more than 300 individual entries to represent the 300 words in Kelly's restricted vocabulary.

It is partly because of this multitude of meanings that some words occur so frequently. In this situation, a system designer has available two courses of action. The designer can choose to allow all the meanings of a word to be placed into computer memory. This will cause the size of the effective vocabulary (and thus the required size of computer memory) to increase several times over the size of the nominal vocabulary. More importantly, such a course of action will require that a way be provided for the program to determine which of a word's multiple meanings is appropriate in a particular situation. The alternative course of action is to allow each word only one meaning for both

the user and the computer. This latter alternative represents an important option and remains to be tested. However, the performance of such a test will require some ingenuity on the experimenter's part. It is fairly likely that, if nothing else, such a situation would increase user dissatisfaction with the system.

Quasi-Natural Language

Some practical application of natural query languages has been accomplished with a program called ELIZA (Weizenbaum, 1966). ELIZA is an elementary natural language system that has been adapted to a few different tasks. Although ELIZA operates in English, it lacks any sophisticated linguistic capabilities. It cannot truly parse or interpret the sentence that is input to it (see Martin, 1973, for criticism of ELIZA's superficialness). However, at the surface level, these limitations are not apparent to the casual user. An example of how ELIZA can be used to assist computer users in the use of system commands is shown below. If a user has difficulty remembering or invoking a system command, he or she can type HELP and initiate a user-computer dialogue. In the example, the user's statements are typed in mixed case and the computer's statements are typed in upper case (Shapiro & Kwasny, 1975, p. 460).

.
.
.

HOW DO YOU DO. WHAT IS YOUR PROBLEM?

?I need some information about files.

FILES MAY BE ASSOCIATED WITH A JOB BY USING EITHER THE ATTACH (DIRECT FILE) OR GET (INDIRECT FILE) COMMANDS. FILES MAY BE MADE PERMANENT BY USING EITHER THE DEFINE (DIRECT FILE) OR SAVE (INDIRECT FILE) COMMANDS.

?I'm not sure if I want direct or indirect files.

A DIRECT ACCESS FILE IS A PERMANENT FILE WHICH, UPON REQUEST, BECOMES ASSOCIATED DIRECTLY WITH THE USER'S JOB. UNLIKE REQUESTS FOR INDIRECT ACCESS FILES, A COPY OF THE PERMANENT FILES IS NOT CREATED FOR USER ACCESS....

.
.
.
.
.
.

?What else can you tell me about indirect access files?

CAN YOU BE MORE SPECIFIC?

?Suppose I want to save indirect files?

THE SAVE COMMAND CREATES......

ELIZA operates by searching input sentences for predefined key words. When a key word is located, it invokes in ELIZA's memory response rules associated to the key word. These rules are used to concatenate phrases from the input sentence with predetermined responses. Shapiro and Kwasny (1975, p. 461) explain that ELIZA follows:

> the general theme that when a command name, a synonym of the command name, or a word implying some use of that command is recognized in a user input, the user is presumed to be asking for information about that command. The initial response is a general description of the usage of the command. . . .
>
> If the same key word reappears, the system responds with more specific information until the feature is completely described. The response to the next use of the keyboard is:
>
> CAN YOU BE MORE SPECIFIC? . . . Further uses of the key word are ignored, allowing less preferred (author's note: i.e., less important) key words to determine the response.

As the example shows, ELIZA is a simple natural query language that is capable of communicating with an untutored user in order to speedily provide instructions on the use of system commands. ELIZA performs no novel manipulation of its data base. Instead, it simply enables the user to locate needed pieces of information quickly. This information could have been found in a command systems manual. However, users might prefer to use ELIZA's interactive dialogue as an instructional aid. Also, the system does not require any special training of the user. It is these aspects of ELIZA which make it of interest to the human factors specialist. Without resorting to an expensive research and development effort, a designer is able to utilize a natural language-like system which has the capability of providing limited services to the user. (ELIZA has also been made to function in other capacities. Weizenbaum, 1966, gives an example of ELIZA functioning as a therapist.)

Debate Over Natural Query Languages

Many researchers feel that, for most purposes, natural language is a poor choice as a query language. Hill (1972) regards English as being too ambiguous for correct interpretation by a computer system. To support this point, Hill presents a number of everyday examples (e.g., "Johnny has grown a foot"). Although statements about the ambiguity of English are correct, it is not obvious that they eliminate English from serving as a query language. Natural query languages, although flawed, already exist (Heidorn, 1976; Waltz, 1976). In an example cited earlier in this paper, the user asks the program PLANES: "How many Phantoms required unscheduled maintenance in April 1975?" The computer understands this question to be about planes and not apparitions. In many instances, a system's limited linguistic capabilities, along with its dedication to a narrow field of knowledge, will prevent irrelevant interpretations of a word. Also, a successful natural query language might feed back to the user a restatement of the command prior to executing it (see discussion below). This would help ensure that the computer understands the statement in the manner intended. The definition of a natural query language need not prohibit learning the limitations of a language through use. The computer may misinterpret a statement such as "Johnny has grown a foot" and therefore produce an absurd

response. The user will then have to reword the query and try again. Learning through experience occurs in all computer languages without destroying their value.

A second line of reasoning pursued by Hill (1972) is that English users frequently do not think through a statement before expressing it. One instance cited is a restaurant menu that lists soups, omelettes, main dishes and then states that "chips and peas included with all the above." After ordering omelettes and getting the bill, the customer learns that chips and peas are not included free of charge with the omelettes (or the soup). The article presents this as an instance where an English statement has failed to accurately explain a situation. However, it is not self-evident that writing the statement in a formal query language would have prevented the error. Many of the examples cited by Hill could just as easily have been misrepresented in a formal computer language as in English. The failure is not due to the language but to the carelessness of its user.

This last point leads to the issue of how a system should respond to queries that it recognizes as faulty. Codd (1974) states that in designing a natural query language, attention must be given to dealing with queries that are poorly conceived. It is not enough for a natural language system to be able to deal with accurate and precise English statements. The system must also be able to clarify ambiguous, imcomplete, or nonsensical statements. This can be done by having the computer initiate a dialogue with the user. The scope of this "clarification" dialogue would be bounded by the data base and by the task objectives of the computer system. The system can help assure that it has correctly interpreted the user's intended meaning by displaying a restatement of the query. This restatement will most likely differ in precision and mode from the user's original formulation. Only after the restatement is accepted by the user does the system proceed to execute the command. (In fact, whether a clarification dialogue is generated or not, all user queries might be checked by having the system formulate and display an internally generated restatement.)

The arguments for and against a natural query language may be summarized as follows. Detractors feel that (1) natural language is too ambiguous to serve as a computer language and (2) when learning to use a formal language, one also learns to formalize the process of problem solving. In other words, using a formal language involves a change in the way one thinks as well as a change in syntax and vocabulary. On the other hand, supporters of natural query languages (Sammett, 1966, 1969) contend that (1) citing examples of natural language ambiguities does not constitute proof that English cannot work as a computer language and (2) natural query languages are not intended to lighten the burden of having to think. Rather, their advantage lies in eliminating the need to remember a host of notational devices which are irrelevant to the problem and which detract from the user's ability to concentrate on the problem per se. In conclusion, the desirability of using English as a computer language has been debated heatedly. However, the evidence presented by both sides has been both anecdotal and inconclusive.

FORMAL QUERY LANGUAGES

Formal query languages, characterized by a highly structured rule system, are an alternative to natural query languages. The division between the two is not distinct. The structure of natural query languages is close to that of English (or whatever the user's own language may be) while the structure of formal query languages is more alien. (An analysis by Moran, 1978, on the syntax of command languages is relevant to the topic of formal language syntax.) However, the differences between formal and natural query languages relate to more than syntax and vocabulary--e.g., the ordering of the particular information within a statement, the presumed default actions, type and arrangement of operands (Miller, 1978). SEQUEL is an example of a formal query language. In this language (Reisner, 1977), the command for "Find all employees who work for Mike Smith and who make less than $20,000" is:

```
SELECT    NAME
FROM      EMP
WHERE     MGR = 'SMITH'
AND       SAL < 20000
```

Although numerous formal query languages already exist, there are no established human factors standards by which these languages can be comparatively evaluated. However, human factors studies have evaluated individual strength and weaknesses within existing formal query language.

Ease of Learning

Some researchers have investigated the ease with which both programmers and nonprogrammers can learn new query languages. In an experiment which compared formal query languages, Greenblatt and Waxman (1978) taught one of three languages (Query by Example, SEQUEL, algebraic language) to college students who had some previous computer training. (Query by Example and SEQUEL, as well as Interactive Query Facility, are query languages packaged by IBM.) The training sessions took less than 2 hours. On testing, the students were able to translate correctly two-thirds of the test questions from English to formal query language. Other experiments (Gould & Ascher, 1975; Reisner, 1977; Thomas & Gould, 1974) with nonprogrammer participants have reported similar success.

Layering

Although the prime objective of most query language research is the evaluation of specific languages, some research has produced more general results. These results are tentative but still useful as guides in an otherwise barren area. Reisner (1977) performed an experimental study of the language SEQUEL. She found a wide range in the ease with which various features of the language were learned. She therefore recommended that the language be treated in a "layered" fashion. "That is, the features should be partitioned into groups,

or layers, with the easier layers intended for users of limited sophistication or need in query writing, the layers increasing in difficulty with the sophistication and needs of the users" (Reisner, 1977, p. 222). Such a recommendation might be valid for any formal query language. In such a layered language, each user could advance to the limit of his or her ability or need. Then, even individuals of limited talent or need would be able to get some use out of the computer system.

Grammatical and Spelling Errors

Reisner (1977) also analyzed the kinds of minor errors made during the writing of query statements. She observed that a large portion of the participants in her experiment made errors of the following types: ending errors (e.g., used "names" for "name," "dispatched" for "dispatch"); spelling errors (despite the fact that the correct spelling was available); and synonym errors (e.g., used "employee" for "personnel," "seniority level" for "seniority"). As a corrective action, Reisner recommended that query languages incorporate computer aids. These aids would include routines which: (1) were capable of matching word stems, (2) corrected spelling errors, and (3) contained a synonym dictionary.

An Army Research Institute (ARI) report by Fields, Maisano, and Marshall (1978) investigated some of these same points. In this experiment, participants typed text into a computer system. The system was programmed to include either: (1) a spelling correction feature, or (2) an autocompletion and "English option" feature. The spelling correction feature operated by comparing unknown terms typed into the computer with terms listed in the program's internal dictionary. It identified that internal term which most closely matched the anomalous term. The found term was presented to the user who then determined if that was the term he or she had meant to input. Autocompletion is a feature which allows users to type in only as much of the initial part of the word (or its code) as is required to uniquely identify it. The program then automatically completes the word for the user. The English option permits users to type in either the English word itself or its established abbreviation (or code). Fields et al. found that their spelling corrector feature reduced the number of spelling errors (that the user would have been otherwise required to correct) by 11%. (It should be noted that the effectiveness of spelling correctors depends upon the state of software technology and not upon operator performance.) On the other hand, when the autocompletion with English option feature was available to the participants, the error rates increased in comparison to a control condition which lacked these features. Although autocompletion was utilized heavily by the participants, the English option was not. Instead, participants showed a strong preference for using codes (typically in conjunction with autocompletion) and rarely did they use words. This experiment indicates that while inexperienced users show a strong preference for the autocompletion option, they experience some difficulty in using it properly. No doubt the benefits derivable from these features depend upon the task being performed and the experience of the operator.

Creating Statements

Gould and Ascher (1975) considered three stages that a user must go through in producing a query statement. First comes the formulation of the problem. Second, the preparation of a plan to solve the problem. And third, the coding of the problem. The authors report that when a query required the establishment of a temporary (i.e., intermediary) variable, the times required for the planning and the coding stages were affected, but not the time required by the formulation stage. In contrast, the research found that when the problem given to a participant was poorly presented, the time required to formulate it increased, while the times required by the other two steps were unaffected.

Semantic Confusion

Gould and Ascher (1975) also found that participants had difficulty with such operations as "or more" and "or less" (e.g., converting the statement "over 50 years old" into "51 or more"). A similar difficulty (e.g., translate "more than 5 years" into "<1969") was observed by Thomas and Gould (1974) in participants working with a natural query language. Both studies also found that participants frequently confused operator's that are semantically similar (e.g., "SUM" and "COUNT"). Thus, there is a need to identify operators which are semantically confusable and to disambiguate them. One method for doing so is through improved training. Another means for reducing the confusion between operators is to give them names which the user will find more distinctive and self-explanatory. Feedback is also useful as a general solution to these and other query language problems. One might devise a feedback feature capable of rephrasing a statement and displaying it back to the user (see earlier discussion). This would occur prior to statement execution and would enable the user to see if the computer understood the statement in the same way as the user intended. The incorporation of such a feature should include a way for experienced operators to turn it off if so desired.

Term Specificity

An ARI report by Potash (1979) investigated the issue of term specificity. This problem is best explained with an example. Imagine a query language designed for accessing personnel files. This language might contain specific retrieval terms such as NAME, AGE, SEX. It also might contain a global term such as NAS (i.e., <u>n</u>ame--<u>a</u>ge--<u>s</u>ex) which retrieves all the information that is retrieved by the three specific terms. Potash investigated the possible benefit of including global terms in a query language. In his experiment, military participants were first instructed on the use of a simplified version of GIM II Query Language. (GIM II was developed by TRW for use on systems, e.g., ASSIST, that they produce.) One group of participants (the "specific group") had only specific terms available to them. A second group of participants (the "global-specific group") used the same specific terms along with a number of global terms. (All of the information retrieved through the use of a single global term could also be obtained by using a number of specific terms.) To assess data entry performance under the two experimental conditions, participants were required to (1) translate a number of English text problems into query statements and (2) enter (i.e., type) the query statements. The two groups showed no differences in the time required to produce the query statements or in the

number of query statements correctly produced. However, the global-specific group saved substantial time in entering (i.e., typing) the query statements. (Statements containing global terms are shorter in length.) Participants also evaluated the availability of global terms as highly preferable. Potash (1979, p. 16) concluded that the "use of global terms is not recommended unless the specific items of information subsumed under the global term are normally retrieved together frequently."

ADDITIONAL QUERY LANGUAGE CONSIDERATIONS

Data Organization

Durding, Becker, and Gould (1977) studied the effects of data organization upon performance. For their experiment, they used sets of word stimuli which had a "natural" organization (i.e., hierarchy, network, list, or table). Participants were given a set of word stimuli and told that the words were related in some way. The participants' task was to discover the relationship and then to rewrite the words so as to make the relationship obvious. With relative ease, participants were able to perform the task. However, when they were instructed to organize the sets into a format that they did not perceive as natural, participants had difficulty in preserving the intrinsic relationships among the words. Although this particular finding is not surprising, the authors used it to make an important point: care must be taken to assure that any organization imposed upon a data base is in accord with the organization perceived as natural by the user. "If the data concern the hierarchical structure of a business, then the user should be able to manipulate the data mentally according to the principles of hierarchical organizations and safely assume and expect that the retrieval system can and will do likewise" (Durding et al., 1977, p. 13). Should the system not be capable of such manipulation, then the user's ability to extract information from it will most likely be impaired.

Codd (1974) also regarded the user's perception of the data base to be of critical importance in properly designing a query language system.

> The user's view of the data in a formatted data base has a fundamental impact on the way he conceives and formulates queries and other types of transactions . . . the [user's] data model [i.e., view of the data] clearly should not have a multiplicity of structural alternatives for representing data. Such a multiplicity is incompatible with the casual user's unwillingness to consciously engage in a learning process and with his tendency to forget what he may have learned unconsciously, because of the irregularity of his interactions. (Codd, 1974, p. 182)

Quantifiers

Another important component of user-machine communication is the use of quantifiers (e.g., "all," "some," "none"). Thomas (1976) reported that users have great difficulty in using quantifiers correctly when formulating query statements. The difficulty with the use of quantifiers is not unique to query languages. Instead, people in many diverse situations appear to have great difficulty in using quantifiers properly (i.e., in the way of logicians).

Thomas reported that people frequently failed to give precisely correct responses when asked to interpret Venn diagrams or to interpret English statements containing quantifiers. Frequently, their responses were either incorrect or consistent with the stimulus without describing it uniquely. Also, there was large variability in performance, as measured both within-subject and between-subjects. It should be noted that all quantifiers were not equally difficult for the individuals to use. The quantifiers "some" and "all" presented much difficulty while the terms "no" and "none" were hardly of any problem.

Consistent with the above experimental findings were observational data which show that in real life dialogue, people rarely use quantification in the logician's sense. Miller and Becker (as reported by Thomas, 1976) note that people more often use qualificational statements (e.g., "Put the red block in the box") than they use quantificational statements ("Given anything which has the property red, and has the property of being a block, that thing also has the property that it belongs in the box") or conditional statements ("If a block is red, then put it in the box"). Thomas goes on to describe three strategies that people use to avoid complex quantification in real life. These strategies are all basically similar in that the subject "homes in" on the desired piece of information rather than asking for it directly. The strategies are:

1. The person engages in a technique similar to the game of 20 questions where he or she asks a series of questions which produce "yes," "no," and "partly" for an answer. The information collected in this manner is used to achieve quantificational disambiguation.

2. Complex sets of relations are not specified in a single concise statement by the individual. Instead, the quantificational information is specified by a sequence of simple statements.

3. Instead of asking for a single complex set of data, the individual requests two or more simpler data sets. The person then proceeds to judge the important set relations among the sets of data.

The observations above clearly indicate that the designer of a query language must be wary of including logical quantifiers which the casual user will not be able to utilize correctly. This caution is equally warranted for a natural language and for a formal language. In both instances, the computer's precise interpretation of a quantificational statement might not coincide with the user's imprecise understanding of logic. Thomas (1976, pp. 16-17) makes the following tentative recommendations about quantification in query languages:

1. Studies should be undertaken concerning the usability of a query system with the particular users and tasks that the system is designed for.

2. Unless one has a logically sophisticated population of users, one should make it possible for users to gather information in ways that are consistent with their natural strategies. Some of the strategies observed above may be fairly universal. The safest course, though, would be to see what strategies particular users may want for a particular system.

3. If, for some reason, a system must use the logician's quantifiers, then a high proportion of errors should be expected and the system designed accordingly. (Intelligible error messages, recovery procedures, etc.)

4. Whenever practical, the human's quantification tasks should be limited to producing or choosing descriptions that are consistent with his needs rather than forcing him to unambiguously specify his needs.

5. Whenever practical, communicate with the user in terms of set identifiers and set disjunctions. (Obviously, in some cases, there is no choice.)

6. A natural language query system should generally not attempt to answer exactly the user's precise question when that question involves quantification. Two users even in the same context may well have in mind by the same string of English words two different set relationships. A more modest and workable strategy--which humans themselves seem to use in communicating with each--is to provide information relevant to the query and satisfying to the user. Note that this strategy does not require that the question answering system induce from the user's question a deep structure corresponding with the user's.

Mixed Initiative Dialogue

A user-computer dialogue can be either computer initiated, user initiated or a combination of the two (mixed initiative). Examples of computer initiated dialogue techniques were given earlier (i.e., question-and-answer, form filling, and menu selection). Query languages can be used in either a computer or user initiated format. Efforts have also been made to develop mixed initiative query systems. An ARI basic research effort along this line is MIQSTURE (Katter, Potash, and Halpin, 1978). In such a system, the user usually leads the dialogue (i.e., user initiated). However, the computer is not a completely passive partner which merely answers questions put forward by the user. Instead, the computer is programmed to take the initiative in the dialogue when it determines that the user has overlooked some aspect of the task or when the user requests computer guidance.

A mixed initiative capability requires that the computer program have some knowledge about the task domain. This is accomplished by programming into the computer a schema or plan of the task. The schema contains information on what factors are important to a particular task and how these factors interrelate. For example, relevant to the task of tank movement are the factors of terrain, enemy positions, weather, obstacles, etc. While a user is querying the computer, the computer may compare the information being requested to the plans and schemata in its memory. In this way the computer can identify the plan or schema appropriate to the user's needs. Then, if the user should fail to request a piece of information relevant to this task, the computer might cue the user to its availability. Thus, users are reminded or made aware of important information that they either have forgotten or did not know existed. In the above example, a commander may query the computer for information about the terrain, enemy positions and obstacles in a given sector. The computer might recognize from these questions that the user is interested in the topic of tank movement. The computer could then ask if the user would also like any

information about the weather. Such a mixed initiative capability is particularly valuable in situations which are characterized by either high stress or information overload. It helps assure that the user will make full use of the computer's potential. However, the development of a workable mixed initiative system still requires much more research before the computer can reliably identify the task domain of the user. (Katter & Bell, 1980, report on an attempt to identify the support features desirable in a military mixed initiative system.)

Two other systems that are related in intent to that of mixed initiative dialogues are worth noting here. They are RITA (Anderson & Gillogy, 1976; Waterman & Jenkins, 1977) and ROSIE (Waterman, Anderson, Hayes-Roth, Klahr, Martin, & Rosenchein, 1979), both of which were developed and are available from the Rand Corporation. (The two systems are rule-based or production systems, i.e., they consist of rules having the form "IF condition THEN action" meaning, if the given condition is true in the current situation then perform the recommended action.) RITA and ROSIE can be used solely as query languages capable of manipulating and retrieving data from a data base. In doing so, they use an English-like structure although they are formal and not natural languages. However, RITA and ROSIE also have more interesting capabilities. Among them is the ability to simulate judgmental or subjective decisions. Thus RITA and ROSIE function as judgmental retrieval systems as well as data retrieval systems. However, the intended use of these two systems is not to have them substitute for human thought. Rather, they provide judgmental evaluations against which analysts can compare their own decision making process. In so doing, the analysts become more conscious and more critical of the complex and ill-defined thought processes involved in reaching a judgmental decision. It is this interaction between the analyst and the computer that leads to an improved decision making process. As in the case of mixed initiative systems, RITA and ROSIE become even more valuable in critical or time-constrained situations (e.g., battlefield situations) where the human decision maker comes under considerable stress. In the same vein as RITA and ROSIE, military systems are being developed to assist battlefield commanders make their decisions (e.g., TACFIRE is an artillery system which suggests fire parameters).

Studies of Person-to-Person Communication

In order to improve the conversational interaction between the user and the computer, research has been performed on person-to-person communications (Chapanis, 1975; Kelly, 1975). A discussion of the similarities and differences between these two forms of communication is offered by Nickerson (1976). To start, Nickerson identifies some of the features that are characteristic of interperson conversations. They include bidirectionality, sense of presence, rules for transfer of control, mixed initiative, etc. He then probes the extent to which these features are or should be incorporated into user-computer interactions. For example, some computerized tasks may be best served with a minimal amount of bidirectionality. In these instances, it is more desirable to have information flow most freely in a single direction. On the other hand, sense of presence (i.e., knowing that the other party is paying attention) is crucial to both interperson and user-computer interactions. In the latter case, users should be assured by the system that their query has been registered and that it is either being processed or being delayed. If this assurance is not readily available, users become frustrated and dissatisfied with the system.

The paper goes on to discuss the appropriateness or inappropriateness to user-computer communication of other features characteristic of interperson conversation. Nickerson (1976, p. 110) concludes his discussion of the conversational nature of user-computer interactions with the following statement:

> ". . . there are two contentious remarks that I would like to make regarding the notion of conversational interaction between persons and computers. The first is that the differences between the person-computer interactions that take place today and interperson conversations are far greater than the similarities between them. The second is that interperson conversation may be, in some respects, an inappropriate and misleading model to use as a goal for person-computer interaction."

The applicability of interperson communication as a model for user-computer interaction will most likely change with the changing state of technology. As interactive systems become more genuinely interactive, some complex aspects of interperson communication will become valuable models for user-computer dialogues. Good examples are the strategies for extracting and consolidating information from a running dialogue (Chapanis, 1975; Thomas, 1978). Information is not always transmitted (i.e., packaged) in the most compact form (for example, see the discussion of quantifiers above). Future interactive systems may be designed to formulate the user's query from a series of user-computer exchanges (i.e., "clarification" dialogue). In these instances, knowledge of interperson communication may be valuable in successfully designing the form of a user-computer dialogue.

Evaluating Specific Features

Much of the research discussed above has been of a broad nature (e.g., quantifiers, the feasibility of natural query languages). However, through experimentation, decisions can also be made about more specific query language options. Sime, Green, and Guest (1973) used experimentation to determine the relative superiority of specific computer language features. (In their paper, Sime et al., compared a nestable construction to a branch-to-label construction.) To ensure that their experimental evaluation of the specific language features was not contaminated by other computer language features (e.g., input/output statements, logic statements), the authors created separate microlanguages, each having no other feature but the feature of interest (i.e., nesting or branching). These microlanguages were then taught to participants and tested for their ease of use. Through this technique, Sime et al. were able to determine which one of the language features was more desirable from a human factors point of view. (The authors reported that nesting was superior to branching.)

Abbreviations

It is common for computer languages to include abbreviations for some of the words in their vocabulary. This is efficient when it allows the user to reduce the number of key strokes required to input a command. However, it becomes frustrating when the user cannot recall an abbreviation which is to be either entered or interpreted. A report by Moses and Potash (1979) performed a series of experiments designed to evaluate the memorability and appeal of

abbreviations formed by the following five techniques: (1) simple truncation; (2) truncation with the second letter also removed; (3) contraction by removal of both the vowels and the letters H, W, and Y (the first letter of the word is never removed); (4) contraction by removal of the highest frequency letters (the first letter of the word is never removed); and, (5) abbreviation according to military standards (the military formed these abbreviations by consensus). The abbreviations formed by each of the above techniques were tested in three manners. First, participants were asked to rate how well each one of the abbreviations represented its corresponding term. Second, the participants were shown an abbreviation and asked to decode it (i.e., produce the original term). Third, the participants were shown a term and asked to encode it (i.e., produce an abbreviation of their own choosing). The tests showed that overall simple truncation performed equal to or better than and was preferred over the other four techniques. It is probably also correct that the technique of simple truncation is both easiest to remember and simplest to apply.

Some words of caution about using abbreviations. First, it might be worthwhile to include an English option feature in the computer program. This feature allows the user to input either the abbreviation or the full English term. (As discussed previously, Fields et al., 1978, tested the English option but in conjunction with autocompletion. The heavy use of autocompletion by the participants in that experiment made it difficult to reach any conclusion about the English option per se.) A second word of caution is that abbreviations should generally not be used for output. Also, abbreviations should be significantly shorter (not just one or two characters) than the original term and they should also be mnemonically meaningful (Engel & Granda, 1975).

SUMMARY

The findings that have been reported in this paper are neither absolute nor definitive; indeed they are rather tentative. Still, they do represent the knowledge gathered to date from human factors research in the area of query languages. For the designer of a query language, this body of knowledge may be the only guidance available. It is thus useful to consolidate the information that has been presented here. Two compendiums of this information are presented. The compendium that appears immediately below is a summary of the information presented in this paper and was written for human factors specialists. The compendium that appears in Appendix B was written in the form of a guideline for system designers. In addition to material that was discussed in this paper, Appendix B contains recommendations that come from Nickerson and Pew (1977). In some operational situations, the recommendations reported may be contraindicated by immediate system requirements.

Summary: General

1. Data Organization--

    a. The organization of a data base should be in accord with what is perceived to be natural by its users (Durding et al., 1977).

    b. The user's perception of a data base should be sufficiently structured so as to enable rapid identification of those parts in which the user is interested (Codd, 1974).

2. Quantifiers (Thomas, 1976)--People have great difficulty in properly using quantifiers (i.e., in the way of logicians).

   a. "Whenever practical, the human's quantification tasks should be limited to producing or choosing descriptions that are consistent with his needs rather than forcing him to unambiguously specify his needs."

   b. "One should make it possible for users to gather information in ways that are consistent with their natural strategies." (These strategies are discussed in the text.)

3. Mixed Initiative Dialogues--are potentially valuable but still require more research (Katter et al., 1978). Systems which aid the user in making subjective or judgmental decisions are also being perfected (Anderson & Gillogy, 1976; Waterman et al., 1979).

4. Person-to-Person Communication--All of the characteristics of inter-person communication are not appropriate to human-computer interactions. The former should be used selectively as a model for the latter (Nickerson, 1976). However, this situation will change as the user-computer dialogue becomes more truly interactive.

5. Evaluating Language Options--One can decide between specific query language options by creating separate microlanguages, each having no other feature but the feature of interest. Performance on these micro-languages can then be experimentally compared in order to decide which option is preferable (Sime et al., 1973).

6. Restatement (Feedback) of User's Query--Prior to the execution of a user's query, the computer should rephrase the query and display it for user acceptance. This assures that the user's intended meaning has been correctly interpreted by the computer (Codd, 1974).

7. Abbreviations--

   a. Simple truncation performs as well or better than other abbreviation techniques (Moses & Potash, 1979).

   b. In general, do not use abbreviations for output (Engel & Granda, 1975).

Summary: Natural Query Languages

Operational natural query languages have been created (Heidorn, 1976; Waltz, 1976) but they are limited in both scope and linguistic capability (Petrick, 1976). In addition, a debate continues over whether natural language is appropriate for use as a computer language (Hill, 1972; Sammett, 1966, 1969).

8. Protocols and Restricted Syntax (Gould et al., 1976)--

   a. People are equally capable of preparing "procedural" protocols (i.e., how-to-instructions) as they are of preparing "description" protocols (merely describing the scene).

b. Limited experimentation has shown that people are able to successfully function with a restricted natural language syntax.

9. Restricted Vocabulary--People are able to successfully function with a restricted vocabulary (and an unrestricted syntax) during person-to-person communication. However, there is an increase in user dissatisfaction and the generality of these results to a user-computer dialogue has not been tested (Kelly, 1975).

10. Clarification Dialogue and Feedback--Attention must be given to dealing with natural language queries that are poorly conceived. In these instances, the system should be capable of conducting a "clarification" dialogue. (Also see statement 6 above.) (Codd, 1974)

11. Quasi-natural Languages, such as ELIZA (Weizenbaum, 1966), may be useful in situations where the system's task is both narrow and well defined. An example of this is a HELP routine prepared by Shapiro and Kwasny (1975).

Summary: Formal Query Languages

A number of investigators (Greenblatt & Waxman, 1978; Gould & Ascher, 1975; Reisner, 1977; Thomas & Gould, 1974) have reported success in training students to use a formal query language in a relatively short time.

12. Layering--The features of a query language "should be partitioned into groups, or layers, with the easier layers intended for users of limited sophistication or need in query writing, the layers increasing in difficulty with the sophistication and needs of the users" (Reisner, 1977).

13. Semantic Confusion--(Gould & Ascher, 1975; Thomas & Gould, 1974)--

a. People have difficulty with such operations as "or more" and "or less" (e.g., converting "over 50 years old" into "51 or more").

b. People frequently confuse operators which are semantically similar (e.g., "SUM" and "COUNT"). Confusion between operators might be reduced by giving them names that are distinctive and self-explanatory or through added emphasis during training.

14. Term Specificity--For inexperienced users, the incorporation of global terms (i.e., terms which subsume a number of specific terms) into a query language increases the speed of data entry (i.e., typing) but does not affect other performance. Therefore, the "use of global terms is not recommended unless the specific items of information subsumed under the global terms are normally retrieved together frequently" (Potash, 1979).

PROSPECTIVE RESEARCH

Many ideas for future research can be derived from the papers that have been discussed here. Three ideas are particularly striking to this author and each one relates to the form and efficiency of query languages.

1. A point is made in this paper that people do not naturally express complex thoughts in a single statement. (See the sections on "Quantifiers" and on "Person-to-Person Communication.") Rather, people tend to break a single, complex thought into a series of simple and redundant statements. For example, most individuals would probably not request the following information in a single statement.

"Give me all reports on units which belong to the same Army group as the 9th Soviet Battalion; and have chemical warfare capability; and were in transit during the last 48 hours or have been observed in sector A in the last 48 hours; and have had either training on desert terrain or have had experience on desert terrain."

It might be preferable, from the user's point of view, to present the above query via a string of statements:

"Give me all reports on units which meet the following conditions. The unit should belong to the same Army groups as the 9th Soviet Battalion. In addition, the unit should have chemical warfare capability. In addition, the unit should have been in transit during the last 48 hours or it should have been observed in sector A in the last 48 hours. In addition, the unit should have had either training on desert terrain or have had prior experience on desert terrain."

Although the above two formats for writing a query statement are only marginally different in style, there could be a significant difference in both user preference and user comprehension. It should be noted that the issue of style being discussed here is relevant to both natural and formal query languages. Indeed, the latter might be more impacted by this issue than the former. Formal query statements are intrinsically alien to the user and thus more prone to misunderstanding.

2.a. This paper discusses the advisability of having the computer restate and feedback the user's query. Only if the user accepts the restatement of the query, is it acted upon. The intent here is to assure that the query is being correctly understood by the computer. However, the potential benefit of such feedback has never been empirically determined. Since it would be costly to develop such a restatement capacity, it seems prudent to establish its value.

b. In conjunction with determining the cost-effectiveness of a restatement capability, one must also determine the alternative ways by which feedback could be accomplished. For example, should the query be restated in a single, continuous statement or should it be broken down into a string of independent statements (e.g., see research question 1 above). The optimal form that a restatement should take is a complete research issue in itself.

3. This paper discusses the possibility of using restricted syntax. For example, a query language might consist of a dictionary of acceptable sentence "skeletons." These sentences could be in English although the language itself is formal. Examples of such skeletons are:

"List all ______________________ which satisfy the following conditions:"
e.g., units, battalions, etc.

"Must be within the same ________________________ as _________________."
e.g., corps, battalions, etc. — i.e., name of unit

"Must have ____ warfare capability."
e.g., chemical, nuclear, psychological, etc.

"Were observed to be in ____________________________________."
e.g., transit, sector A, training, etc.

Each skeleton sentence would include a set of words which can be legally inserted into its blanks. (In the examples above, the words are shown under the blanks.) Also, the sentences could be joined together by "and," "or," and other conjunctions to form a query statement. The feasibility and efficiency of such a query language might be tested. Although the sentences presented above were arbitrarily created, an actual system based on restricted syntax should consist of sentences created in a systematic manner. By understanding the system, users could avoid the burden of having to memorize each individual skeleton. Until the understanding is achieved, novices could still function with the system through the use of job aids which depict the skeleton sentences and the words to be inserted into them.

REFERENCES

Anderson, R.H. and Gillogy, J.J. Rand intelligent terminal agent (RITA): Design philosophy. Santa Monica, CA: Rand Corporation, R-1809-ARPA, February 1976.

Chapanis, A. Interactive human communication. Scientific American, 1975, 232, 36-42.

Codd, E.F. Seven steps to rendezvous with the casual user. In J.W. Klimbie & K.L. Koffeman (Eds.) Data Base Management: Proceedings of the IFIP TC-2 Working Conference on Data Base Management Systems. Amsterdam: North-Holland Publishing Co., 1974.

Durding, B.M., Becker, C.A., and Gould, J.D. Data organization. Human Factors, 1977, 19, 1-14.

Engel, S.E. and Granda, R.E. Guidelines for man/display interfaces. Poughkeepsie, New York: IBM Poughkeepsie Laboratory, Technical Report TR 00.2720, December 1975.

Fields, A.F., Maisano, R.E., and Marshall, C.F. A comparative analysis of methods for tactical data inputting. Alexandria, Virginia: US Army Research Institute for the Behavioral and Social Sciences, Technical Paper 327, September 1978. (NTIS No. AD A060 562).

Gould, J.D. and Ascher, R.N. Use of an IQF-like query language by non-programmers. Yorktown Heights, New York: IBM Thomas J. Watson Research Center, Research Report RC 5279, February 1975.

Gould, J.D., Lewis, C., and Becker, C.A. Writing and following procedural, descriptive, and restricted syntax language instructions. Yorktown Heights, New York: IBM Thomas J. Watson Research Center, Research Report RC 5943, April 1976.

Gram, C. and Hertweck, F. Command languages: Design considerations and basic concepts. In C. Unger (Ed.), Command Languages. Amsterdam: North-Holland Publishing Co., 1975.

Greenblatt, D. and Waxman, J. User oriented query language design. Symposium Proceedings: Human Factors and Computer Sciences. The Human Factors Society Potomac Chapter and Technical Interest Group-Computer Systems, Washington, D.C., June 1978.

Heidorn, G.E. Automatic programming through natural language dialogue: A survey. IBM Journal of Research and Development, 1976, 20, 302-313.

Hill, I.D. Wouldn't it be nice if we could write programs in ordinary English--or would it? The Computer Bulletin, 1972, 16, 306-312.

Katter, R.V. and Bell, G. Experimental evaluation of MIQSTURE: An online interactive language for tactical intelligence processing. Alexandria, Virginia: US Army Research Institute for the Behavioral and Social Sciences, Research Note 80-19, June 1980.

Katter, R.V., Potash, L.M., and Halpin, S.M. MIQSTURE: Design for a mixed initiative structure with task and user related elements. Proceedings of the 22nd Annual Meeting of the Human Factors Society, Detroit, Michigan, October 1978.

Kelly, M.J. Studies in interactive communication: Limited vocabulary natural language dialogue. Baltimore, Maryland: Johns Hopkins University, Dept of Psychology, Technical Report 3, August 1975. (NTIS No. AD A019 198)

Leavenworth, B.M. and Sammet, J.E. An overview of nonprocedural languages. Yorktown Heights, New York: IBM Thomas J. Watson Research Center, Research Report RC 4685, January 1974.

Martin, J. Design of man-computer dialogues. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1973.

Miller, L.A. Behavioral studies of the programming process. Yorktown Heights, New York: IBM Thomas J. Watson Research Center, Research Report RC 7367, October 1978.

Miller, L.A. and Thomas, J.C., Jr. Behavioral issues in the use of interactive systems. Yorktown Heights, New York: IBM Thomas J. Watson Research Center, Research Report RC 6326, December 1976. (Also International Journal of Man-Machine Studies, 1977, 9, 509-536.)

Moran, T.P. Introduction to the command language grammar: A representation for the user interface of interactive computer systems. Palo Alto, Calif.: Xerox Palo Alto Research Center, Report SSL-78-3, AP Memo 111, October 1978.

Moses, F.L. and Potash, L.M. Assessment of abbreviation methods for automated tactical systems. Alexandria, Virginia: US Army Research Institute for the Behavioral and Social Sciences, Technical Report 398, August 1979. (NTIS No. AD A077 840).

Nickerson, R.S. On conversational interactions with computers. In S. Treu (Ed.), User-Oriented Design of Interactive Graphic Systems, Proceeding of ACM/SIGGRAPH Workshop, Pittsburgh, PA, October 14-15, 1976.

Nickerson, R.S. and Pew, R.W. Person-computer interaction. Chapter 6 in The C^3-System User. Vol. 1: A Review of Research on Human Performance as it Relates to the Design and Operation of Command, Control & Communication Systems. Cambridge, Mass.: Bolt Beranek and Newman, Inc., BBN Report No. 3459, February 1977.

Petrick, S.R. On natural language based computer systems. IBM Journal of Research and Development, 1976, 20, 314-325.

Potash, L.M. Effects of retrieval term specificity on information retrieval from computer-based intelligence systems. Alexandria, Virginia: U.S. Army Research Institute for the Behavioral and Social Sciences, Technical Report 379, July 1979. (NTIS No. AD A072 312).

Ramsey, H.R. and Atwood, M.E. Human factors in computer systems: A review of the literature. Englewood, Colorado: Science Applications, Inc., Technical Report SAI-79-111-DEN, September 1979 (NTIS No. AD A075 679).

Ramsey, H.R., Atwood, M.E., and Kirshbaum, P.J. A critically annotated bibliography of the literature of human factors in computer systems. Englewood, Colorado: Science Applications, Inc., Technical Report SAI-78-070-DEN, May 1978. (NTIS No. AD A058 081). (Also JSAS Catalog of Selected Documents in Psychology, 1979, 9, 15. MS. No. 1822).

Reisner, P. Use of psychological experimentation as an aid to development of a query language. IEEE Transactions on Software Engineering, 1977, SE-3, 218-229.

Sammet, J.E. The use of English as a programming language. Communications of the ACM, 1966, 9, 228-230.

Sammet, J.E. Programming languages: History and fundamentals. Englewood Cliffs, New Jersey: Prentice Hall, 1969.

Shapiro, S.C. and Kwasny, S.C. Interactive consulting via natural language. Communications of the ACM, 1975, 18, 459-462.

Sime, M.E., Green, T.R.C., and Guest, D.J. Psychological evaluation of two conditional constructions used in computer languages. International Journal of Man-Machine Studies, 1973, 5, 105-113.

Thomas, J.C. Quantifiers and question-asking. Yorktown Heights, New York: IBM Thomas J. Watson Research Center, Research Report RC 5866, February 1976.

Thomas, J.C., Jr. A design-interpretation analysis of natural English with applications to man-computer interaction. International Journal of Man-Machine Studies, 1978, 10, 651-668.

Thomas, J.C. and Gould, J.D. A psychological study of Query by Example. Yorktown Heights, New York: IBM Thomas J. Watson Research Center, Technical Report RC 5124, November 1974. (Also AFIPS Conference Proceedings, 1975, 44, 439-445).

Waltz, D.L. Natural language access to a large data base. Computers and People, 1976, 25, 19-26.

Waterman, D.A., Anderson, R.H., Hayes-Roth, F., Klahr, P., Martin, G., and Rosenschein, S.J. Design of a rule-oriented system for implementing expertise. Santa Monica, California: Rand Corporation, N-1158-1-ARPA, May 1979.

Waterman, D.A. and Jenkins, B.M. Heuristic modeling using rule-based computer systems. Santa Monica, California: Rand Corporation, P-5811, March 1977.

Weizenbaum, J. ELIZA--a computer program for the study of natural language communications between man and machine. Communications of the ACM, 1966, 9, 36-45.

APPENDIX A

The User-Computer Interface: Other Review Papers

This paper is concerned solely with query languages. However, query languages are but one element in the complex structure of a computer system. A brief description of papers on other issues seems appropriate since the material in the present paper is meant to complement the information presented in these other papers.

Guidelines for Man/Display Interfaces by Engel and Granda (1975) is a useful guideline to software designers interested in human factors issues. Areas covered by the document are: display frame formats (highlighting, data presentation, and screen layout); frame content (feedback to the user, labeling, messages, and interframe considerations); command languages (abbreviations and prompting); recovery procedures; user entry techniques (hardware control methods, entry stacking, implicit prompting); response times and behavioral principles.

Human Factors in Computer Systems: A Review of the Literature by Ramsey and Atwood (1979) and "Person-Computer Interaction" (chapter 6) of The C^3-System User. Vol. 1: A Review of Research on Human Performance as it Relates to the Design and Operation of Command, Control and Communication Systems by Nickerson and Pew (1977) are two documents which present an extensive overview of the field. These documents contain critical discussions of a large number of issues. Each issue is described, commented upon and the principal reference papers and their findings are reported. Both documents contain short discussions of query languages (Ramsey and Atwood, pp. 85-92 and Nickerson and Pew, pp. 291-295) as well as related matters.

A companion document to Ramsey and Atwood (1979) is A Critically Annotated Bibliography of the Literature of Human Factors in Computer Systems by Ramsey, Atwood and Kirshbaum (1978). This bibliography includes a description of and commentary on hundreds of papers. Each of the above documents is extremely useful as a starting point for any investigation into a particular area of human factors research relating to computers.

An introduction to the specific field of user-computer dialogues is given by Martin (1973). His book, Design of Man-Computer Dialogues, is broad in scope and contains a multitude of examples and case histories. In chapter 7, the book presents 23 styles for displaying dialogues.

Another review paper of potential interest to the reader is Behavioral Issues in the Use of Interactive Systems by Miller and Thomas (1976). In addition to being a review article, this document discusses the conceptual issues underlying the study of the use of computers. The authors also put forth suggestions on ways to improve the user-computer interface. Finally, chapter 4 of Behavioral Studies of the Programming Process by Miller (1978) presents a summary of the IBM research program relating to natural language programming and communication.

APPENDIX B

Query Language: A Compendium of Design Recommendations

These recommendations were compiled from the literature review that is presented in the main body of this paper and from additional sources. In some instances, the recommendations that are presented here go beyond what can be empirically substantiated. These recommendations are not to be considered immutable. Instead, they represent the author's opinion as to what guidelines might be thoughtfully offered at the present time to a system designer.

Recommendations: General

Data Organization

1. The organization of the data base that is presented to the users should match the organization perceived to be natural by the users. The users' natural organization can be discovered through experimentation or by survey.

2. Casual users should not be presented with a multitude of models for representing the data base. A single representation of the data base should be sufficient for the total range of user needs. A multiplicity of data base structures only tends to confuse the casual user.

Quantifiers

3. A query language should minimize the use of quantification terms (e.g., "some," "all"). People have great difficulty in using quantifiers unambiguously. Exceptions to this rule are the quantifiers "no" and "none." When quantifiers are required, the system should have the user choose the desired quantification statement from a set of statements that are designed to maximize their distinctiveness.

Evaluating Language Options

4. Test major query language features prior to adopting them. The text of this paper provides a description of experimental procedures that can be used in deciding between alternative design options.

Feedback of the Query

5. Prior to the execution of a user's query, the computer should rephrase the query and display it for user acceptance. This assures that the user's intended meaning has been correctly interpreted by the computer. (Skilled users should be able to suppress this feature if so desired.)

Abbreviations

6. The method of simple truncation should be used in forming abbreviations for terms, e.g., deleting all but the first three to five letters of the words. The value of this technique is markedly increased when it is uniformly applied (with the possible exception of words which have commonly known abbreviations). Allowance must be made for different words resulting in the same abbreviation when truncated. User understanding of how the abbreviations are formed is valuable.

Dialogue Transactions

7. The system's messages to the user should be in a directly usable form and provide prompts or reminders of the current state of transaction development. The user should not have to refer back to previous transactions in order to determine the present states of the system. Lengthy sequences of transactions should be recapped periodically.

8. When the system displays information, "it should be in the form needed at that point even if the format is different from that provided in the data base or [from] when it was originally entered. For example, in a payroll or cost-accounting system salaries may be stored in hourly rates, but if the

current activity requires monthly or yearly rates, the computer should make the required transformation and display accordingly."

9. Users should be able to easily modify a request that is revealed to be incorrect. In particular, they should be able to move backwards through a dialogue sequence in order to change an entry. Introducing such a change should not require re-entry of all the correctly entered material.

10. A small proportion of queries usually accounts for a high proporti of the user's activities. These queries should be designed for greatest ease of accomplishment.

11. Some user queries require a long response time. The computer should acknowledge the receipt of a query and should later indicate that a response is available.

Specific Recommendations: Formal Query Languages

Layering

12. The features of a query language should be partitioned into groups or layers. The easiest layer should be able to stand alone and is intended for users of limited sophistication or limited need. The layers should then increase in complexity for use by more sophisticated personnel. Such a procedure will broaden the base of users.

Semantic Confusion

13. Avoid the use of operators such as "or more" and "or less" (e.g., do not require the user to convert "over 50 years old" into "51 or more"). People have difficulty using these operators correctly.

14. Query language operators should not be given semantically similar names (e.g., "SUM" and "COUNT"). To avoid confusion, operators should be given names that are distinctive and self-explanatory.

Term Specificity

15. For inexperienced users, the use of global terms (e.g., general terms which subsume a number of specific terms) is not recommended unless the specific terms of information subsumed under the global terms are retrieved together frequently. The availability of global terms does increase the speed of data entry (i.e., typing) but does not affect accuracy.

Specific Recommendations: Natural Query Languages

Clarification Dialogue

16. Natural query language systems should be capable of carrying out a "clarification dialogue." Users will frequently input poorly stated queries and it is not sufficient for the system to simply reject them. Instead, the system should be capable of guiding the user through a dialogue which will result in the formulation of a proper statement.

Quasi-Natural Languages

17. Quasi-natural languages should be considered as design options in situations where it is neither possible to teach a formal query language to potential users nor is it feasible to develop a natural query language. Quasi-natural languages are English-like in structure but they are not capable of truly "understanding" the text's meaning. For a quasi-natural language to be applicable, the system's task should be narrow and well defined. Examples of the use of a quasi-natural language are given in the text.

DISTRIBUTION

1 US ARMY WESTERN COMMAND ATTN: APPE
1 DEPARTMENT OF THE NAVY TRAINING ANALYSIS AND EVALUATION GROUP
1 HQDA ATTN: DAAG-ED
1 HQ, TCATA ATTN: ATCAT-OP-Q
2 HQDA RESEARCH AND STUDIES OFC
1 MILITARY OCCUPATIONAL DEVELOPMENT DIV DAPC-MSP-O, RM 852C HOFFMAN BLDG 1
4 OASD (MRA AND L)
1 HQDA ATTN: DAMO-RQR
1 HQ TCATA TECHNICAL LIBRARY
1 HQDA ODCSPER
1 USRADCO, STC
1 HQDA ATTN: DAMI-ISI
1 USA AVIATION SYSTEMS COMD ATTN: DRSAV-ZDR
1 USA CORADCOM ATTN: AMSEL-PA-RH
1 USA ARRADCOM ATTN: ATFE-LO-AC
1 HEADQUARTERS, US MARINE CORPS ATTN: CODE MPI-20
2 US ARMY EUROPE AND SEVENTH ARMY
1 1ST INFANTRY DIVISION AND FT. RILEY ATTN: AFZN-DPT-T
1 USA INTELLIGENCE AND SECURITY COMMAND ATTN: IAOPS-TNG-T
2 HQ TRADOC TECHNICAL LIBRARY
1 NAVAL TRAINING EQUIPMENT CEN ATTN: TECHNICAL LIBRARY
1 MILITARY OCCUPATIONAL DEVELOPMENT DIRECTORATE ATTN: ATZI-NCR-MS-M, RM 3N33 HOFFMAN BLDG II
1 DATA ANALYSIS DIVISION ATTN: ATZI-NCR-MD, HOFFMAN BLDG II
1 USA MILPERCEN ATTN: DAPC-POO-T
1 USAFACFA CHIEF, ORGANIZATIONAL EFFECTIVENESS BRANCH
1 8TH INFANTRY DIVISION
1 HQDA ARMY FORCE MODERNIZATION COORDINATION OFFICE
1 NAVAL AIR SYSTEM COMMAND /
1 DCSOPS (DIST 4) ATTN: DAMO-RQI
1 123D USARCOM RESERVE CENTER
1 US ARMY SOLDIER SUPPORT CENTER /
1 DIRECTORATE OF ARMOR AVIATION ATTN: ATZK-AAD
1 USAARMC + FT. KNOX AVIATION DIVISION
1 USA FORCES COMMAND AFIN - DEPUTY C OF S FOR INTELLIGENCE
1 USA FORCES COMMAND AFOP - DEPUTY CHIEF OF STAFF FOR OPERATIONS
1 US ARMY AIR DEFENSE SCHOOL ATTN: ATSA-DTD
1 DIRECTORATE OF TRAINING ATTN: ATZQ-T
1 DIRECTORATE OF COMBAT DEVELOPMENTS ATTN: ATZQ-D
1 HQDARCOM MARINE CORPS LIAISON OFC
1 DEPARTMENT OF THE ARMY US ARMY INTELLIGENCE + SECURITY COMMAND
1 US ARMY SAFETY CENTER ATTN: LIBRARIAN, BLDG 4905
1 USA MISSILE COMMAND ATTN: DRSMI-NTN
1 US ARMY CECOM ATTN: DRSEL-ATDD
1 USA FORCES COMMAND
1 PM TRADE /
1 US MILITARY DISTRICT OF WASHINGTON OFC OF EQUAL OPPORTUNITY
1 NAVAL CIVILIAN PERSONNEL COMD SOUTHERN FLD DIV
22 ARI LIAISON OFFICE
1 7TH ARMY TRAINING COMMAND
1 HQ USAREUR ATTN: DCSOPS
1 HQDA, DCS STUDY OFFICE
1 U.S. NAVY TRAINING ANALYSIS EVALUATION GROUP
1 USACDEC ATTN: ATEC-EX-E HUMAN FACTORS
1 USAFAGOS/TAC SENIOR ARMY ADVISOR
1 INTER-UNIV SEMINAR ON ARMED FORCES + SOC
1 USA ELECTRONIC PROVING GROUND ATTN: STEEP-MT-ES
1 OASA (RDA) DEPUTY FOR SCIENCE AND TECHNOLOGY
1 OFC OF NAVAL RESEARCH /
1 AFHRL/LRT
1 AFHRL/LRLG
1 AIR FORCE HUMAN RESOURCES LAB ATTN: AFHRL/TSR

1 FEDERAL AVIATION ADMINISTRATION CENTRAL REGION LIBRARY, ACE-66
1 AFAMRL/BB
1 AFAMRL/HE
1 NAVAL PERSONNEL R AND D CENTER COMMAND AND SUPPORT SYSTEMS
1 NAVY PERSONNEL R AND D CENTER /
1 NAVY PERSONNEL R AND D CENTER DIRECTOR OF PROGRAMS
1 NAVY PERSONNEL R AND D CENTER /
1 US ARMY AVN ENGINEERING FLIGHT ACTIVITY ATTN: DAVTE-TD
2 OFC OF NAVAL RESEARCH PERSONNEL AND TRAINING RESEARCH PROGRAMS
1 NAVAL PERSONNEL R + D CENTER /
1 OFC OF NAVAL RESEARCH PROJECT OFFICER, ENVIRONMENTAL PHYSIOLOGY
1 DEPT. OF NATIONAL DEFENCE DEFENCE AND CIVIL INSTITUTE OF ENVIR MED
1 NAVAL AEROSPACE MEDICAL RSCH LAB AEROSPACE PSYCHOLOGY DEPARTMENT
1 USA TRADOC SYSTEMS ANALYSIS ACTIVITY ATTN: ATAA-TCA
1 HEADQUARTERS, COAST GUARD CHIEF, PSYCHOLOGICAL RSCH BR
1 USA RESEARCH AND TECHNOLOGY LAB /
1 USA ENGINEER TOPOGRAPHIC LABS ATTN: ETL-GSL
1 USA ENGINEER TOPOGRAPHIC LABS ATTN: STINFO CENTER
1 USA ENGINEER TOPOGRAPHIC LABS ATTN: ETL-TD-S
1 USA MOBILITY EQUIPMENT R AND D COMD ATTN: DRDME-TQ (SCHOOL)
1 NIGHT VISION LAB ATTN: DRSEL-NV-SDD
1 USA TRAINING BOARD ATTN: ATTG-ATB-TA
1 USA HUMAN ENGINEERING LAB
1 USAHEL LIAISON REP, USAAVNC /
1 USA MATERIEL SYSTEMS ANALYSIS ACTIVITY ATTN: DRXSY-C
1 USA RESEARCH OFC /
1 NAFEC HUMAN ENGINEERING BRANCH
1 BATTELLE-COLUMBUS LABORATORIES TACTICAL TECHNICAL OFC
1 USA ARCTIC TEST CEN ATTN: AMSTE-PL-TS
1 USA COLD REGIONS TEST CEN ATTN: STECR-OP
1 USA CONCEPTS ANALYSIS AGCY ATTN: CSCA-RQP
1 USA CONCEPTS ANALYSIS AGCY ATTN: CSCA-JF
1 HQ WRAIR DIV OF NEUROPSYCHIATRY
1 USACACDA ATTN: ATZL-CAC-IC
1 USACACDA ATTN: ATZL-CAC-IM
1 USACAC ATTN: ATZL-CAC-IA
1 USACACDA ATTN: ATZL-CAC-A
1 USA ELECTRONIC WARFARE LAB CHIEF, INTELLIGENCE MATER DEVEL + SUPP OFF
1 USA RSCH DEVEL + STANDARDIZA GP, U.K.
1 USA RESEARCH AND DEVELOPMENT LABS CHIEF, BEHAV SCIENCES DIV, FOOD SCI LAB
1 TRAJANA ATTN: SAJS-OR
1 NAVAL AIR SYSTEMS COMMAND ATTN: AIR-5313
1 ECOM ATTN: AMSEL-CT-O
1 USACDEC TECHNICAL INFORMATION CENTER
1 USAARL LIBRARY
1 HUMAN RESOURCES RSCH ORG (HUMRRO) /
1 SEVILLE RESEARCH CORPORATION
1 USA TRADOC SYSTEMS ANALYSIS ACTIVITY ATTN: ATAA-SL (TECH LIBRARY)
1 UNIFORMED SERVICES UNIT OF THE HEALTH SCI DEPARTMENT OF PSYCHIATRY
1 USA COMPUTER SYSTEMS COMMAND ATTN: COMMAND TECHNICAL LIBRARY H-9
1 HUMAN RESOURCES RSCH ORG (HUMRRO)
1 HUMRRO LIBRARY
1 EUSTIS DIRECTORATE, USAAMRDL TECHNICAL LIBRARY
1 RAND CORPORATION /
1 RAND CORPORATION ATTN: LIBRARY D
1 FEDERAL AVIATION ADMINISTRATION ATTN: CAMI LIBRARY ACC-44D1
1 NAFEC LIBRARY, ANA-64
1 GRONINGER LIBRARY ATTN: ATZF-RS-L BLDG 1313
1 CENTER FOR NAVAL ANALYSIS
1 NAVAL HEALTH RSCH CEN LIBRARY
1 NAVAL ELECTRONICS LAB ATTN: RESEARCH LIBRARY
1 NAVAL PERSONNEL R AND D CEN LIBRARY ATTN: CODE P106

1 AIR FORCE HUMAN RESOURCES LAB ATTN: AFHRL/OTS
1 HQ, FT. HUACHUCA ATTN: TECH REF DIV
1 USA ACADEMY OF HEALTH SCIENCES STIMSON LIBRARY (DOCUMENTS)
1 SCHOOL OF SYSTEMS AND LOGISTICS /
1 USAMERDC TECHNICAL LIBRARY
1 DEPARTMENT OF THE NAVY TRAINING ANALYSIS AND EVALUATION GP
1 NATIONAL CENTER FOR HEALTH STATISTICS /
1 USMA DEPT OF BEHAVIORAL SCI AND LEADERSHIP
1 OLD DOMINION UNIVERSITY PERFORMANCE ASSESSMENT LABORATORY
1 USA COMMAND AND GENERAL STAFF COLLEGE ATTN: LIBRARY
1 USA TRANSPORTATION SCHOOL USA TRANSP TECH INFO AND RSCH CEN
1 NASA HQ /
1 NMRDC PROGRAM MANAGER FOR HUMAN PERFORMANCE
1 NAVAL MEDICAL R AND D COMMAND (44)
1 USA ADMINCEN TECHNICAL RESEARCH BRANCH LIBRARY
2 HQDA USA MED RSCH AND DEVEL COMMAND
1 USA FIELD ARTY BD /
1 NAT CLEARINGHOUSE FOR MENTAL HEALTH INFO PARKLAWN BLDG
1 U OF TEXAS CEN FOR COMMUNICATION RSCH
1 INSTITUTE FOR DEFENSE ANALYSES
1 USA TRAINING SUPPORT CENTER ATTN: ATIC-DST-PA
1 AFHRL TECHNOLOGY OFC (H)
1 PURDUE UNIV DEPT OF PSYCHOLOGICAL SCIENCES
1 USA MOBILITY EQUIPMENT R AND D COMMAND ATTN: DRDME-ZG
1 HQ, USA MDW ATTN: ANPE-OE
1 DA US ARMY RETRAINING BDE RESEARCH + EVALUATION DIVISION
1 CALSPAN HUMAN FACTORS AND TRAINING CENTER
1 USA AEROMEDICAL RESEARCH LAB SCIENTIFIC INFORMATION CENTER
1 USAF SCHOOL OF AEROSPACE MEDICINE AEROMEDICAL LIBRARY (TSK-4)
1 US MILITARY ACADEMY DEPT. OF HISTORY, BLDG 601
1 USA INTELLIGENCE CEN AND SCH ATTN: SCHOOL LIBRARY
1 USA INTELLIGENCE CEN AND SCH ATTN: ATSI-DP
1 MARINE CORPS INSTITUTE
1 NAVAL SAFETY CENTER /
1 USAAVNC AND FT. RUCKER ATTN: ATZQ-ES
1 US ARMY AVN TNG LIBRARY ATTN: CHIEF LIBRARIAN
1 USAAVNC ATTN: ATZQ-D
1 US MILITARY ACADEMY DIRECTOR OF INSTITUTIONAL RSCH
1 USA AIR DEFENSE SCHOOL ATTN: ATSA-CD-MS
1 USAADS-LIBRARY-DOCUMENTS
1 USA AIR DEFENSE BOARD ATTN: FILES REPOSITORY
1 USA INFANTRY BOARD ATTN: ATZB-IB-AE
1 USA INTELLIGENCE CEN AND SCH ATTN: ATSI-DT-SFL
1 USA ORDNANCE CEN AND SCH ATTN: ATSL-TD-TAC
1 USA ARMOR SCHOOL ATTN: ATZK-TD
1 USA ARMOR CENTER DIRECTORATE OF COMBAT DEVELOPMENTS
1 NAVAL POSTGRADUATE SCH ATTN: DUDLEY KNOX LIBRARY (CODE 1424)
1 USA TRANSPORTATION SCHOOL DEPUTY ASST. COMMANDANT EDUCA. TECHNOLOGY
1 USA SIGNAL SCHOOL AND FT. GORDON ATTN: ATZH-ET
1 USA ARMOR CENTER + FT. KNOX OFFICE OF ARMOR FORCE MGT + STANDARDIZATION
1 CHIEF OF NAVAL EDUCATION AND TNG /
1 USA SIGNAL SCHOOL + FT. GORDON EDUCATIONAL TECHNOLOGY DIVISION
1 HQ ATC/XPTD TRAINING SYSTEMS DEVELOPMENT
5 USA INTELLIGENCE CEN AND SCH ATTN: ATSI-ERM
1 US ARMY ARMOR CENTER ATTN: ATZK-TD-PMO
1 USA QUARTERMASTER SCHOOL DIRECTORATE OF TRAINING DEVELOPMENTS
1 US COAST GUARD ACADEMY /
1 USA TRANSPORTATION SCHOOL DIRECTORATE OF TRAINING + DOCTRINE
1 USA INFANTRY SCHOOL LIBRARY /
1 USA INFANTRY SCHOOL ATTN: ATSH-I-V
1 US ARMY INFANTRY SCHOOL ATTN: ATSH-CD
1 USA INFANTRY SCHOOL ATTN: ATSH-DOT-LRD

1 USA INFANTRY SCHOOL ATTN: ATSH-EV
1 USA MP + CHEM SCH/TNG CEN + FT. MCCLELLAN ATTN: ATZN-PTS
1 USA MP + CHEM SCH/TNG CEN + FT. MCCLELLAN DIR, COMBAT DEVELOPMENT
1 USA MP + CHEM SCH/TNG CEN + FT. MCCLELLAN DIR. TRAINING DEVELOPMENT
1 USA MP + CHEM SCH/TNG CEN + FT. MCCLELLAN ATTN: ATZN-MP-ACE
1 USA INSTITUTE OF ADMINISTRATION ATTN: RESIDENT TRAINING MANAGEMENT
1 USA FIELD ARTILLERY SCHOOL MORRIS SWETT LIBRARY
1 USA INSTITUTE OF ADMINISTRATION ACADEMIC LIBRARY
1 USA WAR COLLEGE ATTN: LIBRARY
1 USA ENGINEER SCHOOL LIBRARY AND LEARNING RESOURCES CENTER
1 USA ARMOR SCHOOL (USARMS) ATTN: LIBRARY
1 ORGANIZATIONAL EFFECTIVENESS CEN + SCH ATTN: LIBRARIAN
1 US ARMY INTELLIGENCE CENTER + SCHOOL ATTN: ATSI-TP
1 US ARMY INTELLIGENCE CENTER + SCHOOL ATTN: ATSI-RM-M
1 US ARMY INTELLIGENCE CENTER + SCHOOL ATTN: ATSI-TD-PM
1 US ARMY INTELLIGENCE CENTER + SCHOOL ATTN: ATSI-CD-CS
1 US ARMY INTELLIGENCE CENTER + SCHOOL ATTN: ATSI-ES
1 DEPARTMENT OF THE AIR FORCE AIR UNIVERSITY LIBRARY (ATC)
1 HQ TRADOC TRAINING DEVELOPMENT INSTITUTE
2 BRITISH EMBASSY BRITISH DEFENCE STAFF
2 CANADIAN JOINT STAFF
1 CDLS (W) LIBRARY
1 FRENCH ARMY ATTACHE
1 AUSTRIAN EMBASSY DEFENSE, MILITARY AND AIR ATTACHE
3 CANADIAN DEFENCE LIAISON STAFF ATTN: COUNSELLOR, DEFENCE R AND D
1 ROYAL NETHERLANDS EMBASSY MILITARY ATTACHE
1 CANADIAN FORCES BASE CORNWALLIS ATTN: PERSONNEL SELECTION
2 CANADIAN FORCES PERSONNEL APPL RSCH UNIT
1 ARMY PERSONNEL RESEARCH ESTABLISHMENT
1 NETHERLANDS EMBASSY OFFICE OF THE AIR ATTACHE
6 LIBRARY OF CONGRESS EXCHANGE AND GIFT DIV
1 DEFENSE TECHNICAL INFORMATION CEN ATTN: DTIC-DDA-2
140 LIBRARY OF CONGRESS UNIT DOCUMENTS EXPEDITING PROJECT
1 US GOVERNMENT PRINTING OFC LIBRARY, PUBLIC DOCUMENTS DEPARTMENT
1 US GOVERNMENT PRINTING OFC LIBRARY AND STATUTORY, LIB DIV (SLL)
1 THE ARMY LIBRARY ATTN: ARMY STUDIES SEC
3 / /

NUMBER OF ADDRESSEES 222

TOTAL NUMBER OF COPIES 406